Dynamic Key Ring Update Mechanism for Mobile Wireless Sensor Networks

Merve Şahin
Sabancı University
Istanbul, Turkey
mervesahin@sabanciuniv.edu

Albert Levi
Sabancı University
Istanbul, Turkey
levi@sabanciuniv.edu

***Abstract*—Key distribution is an important issue to provide security in Wireless Sensor Networks (WSNs). Many of the key pre-distribution schemes proposed for static WSNs perform poorly when they are applied to Mobile Wireless Sensor Networks (MWSNs). In this paper, we propose Dynamic Key Ring Update (DKRU) mechanism for MWSNs. The aim of DKRU mechanism is to enable sensor nodes to update their key rings periodically during movement, by observing the frequent keys in their neighbors. Our mechanism can be used together with different key pre-distribution schemes and it helps to increase the performance of them. For the performance evaluation basis, we used our mechanism together with a location based key pre-distribution scheme. Our results show that DKRU mechanism increases the local and global connectivity when it is applied to MWSNs. Moreover, our mechanism does not cause a significant degradation in network resiliency.**

***Index Terms*—mobile wireless sensor networks, key ring update, security, resiliency, connectivity**

I. Introduction

Wireless Sensor Networks (WSNs), consisting of small, autonomous devices called sensor nodes, have increasing range of application areas such as military surveillance, environmental tracking, patient monitoring and smart home applications [1]. All these applications convey sensitive data, so they require a secure communication medium among the sensor nodes and the base station (sink node), where the data is collected. However, sensor nodes have many limitations that make it complicated to develop security protocols for WSNs.

A promising solution on key distribution, which is suitable for most of the requirements and limitations of WSNs, is proposed by Eschenauer and Gligor [2] in 2002. This scheme is based on the notion of the pre-distribution of keying material. It will be referred as the *basic scheme* throughout the paper. Many studies in the literature are based on this basic notion such as the matrix based, polynomial based, combinatorial design based and location based approaches [3].

All these solutions mentioned for the key distribution problem in WSNs assumes that the sensor nodes are static. However, many applications areas of WSNs require the sensor nodes to be mobile. Our unpublished initial analysis shows that random and location based key distribution schemes perform poorly in Mobile Wireless Sensor Networks (MWSNs). Unfortunately, there is an important gap in the literature for the key distribution in MWSNs.

The aim of this study is to turn the node mobility into advantage by providing a smart key ring update mechanism for sensor nodes. Using this mechanism, sensor nodes can re-organize their key rings with the help of the base stations in the area. This mechanism can be used together with different key pre-distribution schemes. Regardless of the initial key pre-distribution scheme, our mechanism increases the local and global connectivity values, without an important decrease in resiliency. Moreover, it does not require an increase in the key ring size and it causes only a small amount of communication overhead. In this paper, we use a deployment knowledge based scheme proposed in [4] for the key pre-distribution. Then we apply our dynamic key ring update (DKRU) mechanism and measure the global connectivity, local connectivity, resiliency and communication overhead of the network via simulations. According to our simulation results, DKRU mechanism provides almost perfect global connectivity and increases the local connectivity by almost 40%, without a significant change in resiliency and communication overhead.

The rest of the paper is organized as follows. In Section II, related work about key distribution in WSNs and MWSNs are summarized. Section III gives background information about the mobility models appropriate for MWSNs. The proposed mechanism is explained in Section IV. Section V presents the performance evaluation of the proposed scheme comparatively and Section VI concludes the paper.

II. Related Work

The scheme proposed in [2] (also called the basic scheme) is composed of three phases. In the key pre-distribution phase, a random set of keys are chosen from a large key pool and loaded to the memory of each sensor node, together with the key identifiers (IDs). These keys form the key ring of the node. After deployment, shared key discovery phase starts. In this phase, sensor nodes broadcast their key identifiers in clear text. If two nodes are in communication range of each other and if they share at least one common key, then they can communicate securely using symmetric encryption. If a pair of nodes does not share any common keys, they are provided with a path key in the path key establishment phase.

The disadvantage of basic scheme is that it brings a tradeoff between connectivity and security. As the key ring size increases, the probability of forming a secure link between two

This work was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under grant 110E180.

978-1-4799-0857-8/13/$31.00 ©2013 IEEE

nodes also increases. However, the network becomes less resilient to node capture attacks. To strengthen the security of basic scheme, different methods are proposed such as [5] and [6]. In q-composite random key pre-distribution scheme [5], two nodes are required to share at least q common keys to form a secure link. Moreover, the communication key is generated as the hash of all shared keys between these two nodes.

To achieve better connectivity and resiliency than the random key pre-distribution schemes, some of the studies use other information such as the deployment location of sensor nodes. The scheme proposed by Du et al. [4] (will be referred as Du's scheme) utilizes the fact that sensor nodes will be deployed as groups, so this deployment knowledge can be used to give common keys only to the neighboring groups, thus increasing connectivity. In this scheme, sensor nodes are divided into groups, and a key pool is designed for each group. The key pools of horizontally, vertically or diagonally neighboring groups have certain amounts of overlapping keys. However, two non-neighboring key pools do not share any key. Groups of nodes are deployed with grid pattern and deployment points follow a two dimensional Gaussian distribution within each grid cell. Then, basic scheme is applied within each group.

The problem with location based schemes is that, when they are applied to MWSNs, usage of deployment knowledge becomes a disadvantage as time progresses. In [7], it is showed that the location based schemes do not have any superiority over random key pre-distribution schemes regarding the MWSNs. Moreover, for certain mobility models, location based schemes may perform far worse than the probabilistic schemes.

Although there is limited work in literature for the key distribution problem in MWSNs, some schemes designed for static networks can be applied to mobile networks to some extent such as [8] and [9]. The approach proposed [10] uses the post deployment knowledge of sensor nodes to prioritize their keys in MWSNs. This study requires the existing of a location finding system and high amount of additional memory to achieve a reasonable connectivity level. The scheme proposed in [11] uses mobile base stations operating as key distribution centers. This scheme is perfectly resilient to node capture attacks, because each node pair uses a different key, generated and distributed by the base station.

III. Mobility models

The survey by Camp, Boleng and Davies [12] is one of the most important studies on WSN mobility models in literature. This study concludes that performance of an ad hoc network can vary significantly with different mobility models. Also, during the performance evaluations, chosen mobility model should closely match the expected real-world scenario. Considering these conclusions, we chose the Random Walk Mobility Model for entity based mobility and the Reference Point Group Mobility Model (RPGM) for group based mobility in our simulations. In Random Walk Mobility Model, nodes randomly choose a direction and speed from pre-defined ranges [12]. They move in that direction for a constant travel time or a constant distance, and then choose a different direction and speed. In RPGM model, a node is chosen as a logical center within each group. Group center chooses a random direction and speed and starts moving to the destination. Other nodes move to a randomly chosen point, which is in a pre-defined radius of the group center.

IV. Proposed Dynamic Key Ring Update Mechanism

In this section, we present our Dynamic Key Ring Update (DKRU) mechanism for mobile wireless sensor networks. Our mechanism can be used together with different key pre-distribution schemes and it can be considered as an extension to the shared key discovery phase. The main purpose of our mechanism is to enable a sensor node to periodically update its key ring according to its neighbors. After each time the shared key discovery phase is performed, a node determines on a set of keys which are frequent among its neighbors, and requests the transmission of these keys from a base station. As a result, during the next shared key discovery phase, the probability of sharing common keys with neighbors increases for each node.

Before a more detailed explanation of our mechanism, the list of symbols we use and the pseudo code of our mechanism is provided in Table I and Fig. 1 respectively. The application process of DKRU mechanism can be examined in five steps for better explanation. These steps will be explained in reference to the pseudo code (Fig. 1).

A. Key Pre-distribution and Deployment

In this study, we used Du's scheme [4] together with q-composite scheme [5] as the key pre-distribution basis for sensor nodes. Sensor nodes are divided into equally-sized groups and a group key pool is prepared for each group. Keys in group key pools are selected from a global key pool, considering the neighboring relations of groups after deployment. Then, a certain number of keys (m) are distributed randomly to each sensor node, from the related group key pool. q value for the q-composite scheme is set to 2, which means at least 2 common keys are required for secure communication of two nodes. In addition, base stations share pre-loaded pair wise keys with each sensor node and they store all the keys of the global key pool in their memory. The pair wise key between node i and a base station is denoted as K_{i-BS}.

After the key pre-distribution phase, nodes and base stations are deployed. As in Du's scheme, grid pattern is used in deployment. At each grid cell, a node group is deployed following a two dimensional Gaussian distribution. The center of each grid cell becomes the deployment point. This part covers the steps 1 to 4 in our pseudo code.

B. Forming the Key Transfer List

After deployment, sensor nodes try to communicate by performing the shared key discovery phase periodically. In shared key discovery phase, sensor nodes broadcast the key IDs in their key rings to see if they share any common keys with their neighbors. Consequently, a node learns the IDs of all keys that exist in its neighbors' key rings. Using this information, a node can easily calculate the frequency of each key that is found in its neighbors' key rings, but not found in

TABLE I. LIST OF SYMBOLS USED IN OUR DKRU MECHANISM

n_i	Sensor node *i*
K_{i-BS}	Pairwise *key shared between node i and base station (BS)*
C_i	List of the Most Frequent Keys belonging to node *i*
T_i	Key Transfer List belonging to node *i*
R_i	List of Remembered Keys belonging to node *i*
m	Size of the key ring
q	Minimum number of common keys required for two neighboring nodes to establish a secure communication (a parameter for q-composite scheme)
tc	Number of frequent key IDs added to Key Transfer List from 1-hop neighbors
p	Probability for adding a frequent key ID to Key Transfer List
t_{max}	Maximum number of keys that a sensor node can transfer from the base station at one time (Maximum Transfer Count)
nc	Node connectivity threshold for key transfer decision
rc	Maximum size for List of Remembered Keys
uc	Usage count threshold for deletion of keys

its own key ring. The IDs of these keys constitute the List of the Most Frequent Keys (C_i) for this node. Then, these frequencies are sorted in decreasing order. Starting with the most frequent key, a node selects tc number of keys for its Key Transfer List (T_i). Each key is selected with a probability of p. In this initial state, T_i list consists of the frequent keys that are found in n_i's 1-hop neighbors. This part corresponds to steps 6 and 7 in our pseudo code.

After nodes establish their initial Key Transfer Lists, they broadcast these lists to their neighbors. In this way, nodes can learn the frequent keys found in their 2-hop neighbors. Nodes have a high probability of meeting with their 2-hop neighbors in the future steps, so this broadcast operation can be considered as an investment in the future. The IDs of unique frequent keys coming from the 2-hop neighbors are added to the T_i list.

At this point, number of key IDs in T_i list may be more than the allowed Maximum Transfer Count (t_{max}). In this case, some of these key IDs are deleted randomly, until the Maximum Transfer Count is reached. The reason for adding randomness to the process of forming Key Transfer List is to prevent the transfer of same set of keys repeatedly. If the transfer lists become repetitive, many of the links are secured by the same set of keys, which will deteriorate the resiliency of network. Another precaution against repetitive transfer lists is to have a List of Remembered Keys (R_i) in each node. While forming the T_i list, the keys in R_i list are also checked and these keys are certainly excluded from T_i list. The detailed usage of R_i list will be explained in the next subsection. The steps 8, 9 and 10 in our pseudo code corresponds the process of finalizing the Key Transfer List for each node.

C. Deciding on Key Transfer

After a node forms its Key Transfer List, it decides whether it needs to transfer these keys or not, according to its *node connectivity*. *Node connectivity* is the ratio of number of neighbors with which a node shares common keys to the number of all neighbors. This ratio can easily be calculated at the end of the shared key discovery phase. If the connectivity of a node is less than a threshold value (nc), this node requests the transfer of new keys from the base station. However, if connectivity of a node is greater than the nc threshold, it does not transfer any keys. Instead, the key IDs in its Key Transfer List (T_i) are added to the List of Remembered Keys (R_i). Node remembers these keys because when forming the Key Transfer List next time, these keys are excluded from the possibility of transfer. The purpose of List of Remembered Keys is again to prevent the transmission of same set of keys repeatedly. If size of the R_i list has already reached its maximum value (rc), then enough number of keys are deleted from R_i list, starting with the oldest ones. In this way, the latest remembered keys are prioritized. This part covers the step 12 in our pseudo code, excluding 12.a.i and 12.a.ii, which will be explained in following subsections.

D. Key Deletion Process

Another property of our mechanism is that the size of the key ring of a node never exceeds the predefined key ring size m. Before a node transfers new keys, it deletes the required number of existing keys. Key deletion process has two steps. For the first step, each node stores *key usage count* values for all of its keys. *Key usage count* is calculated as the number of times a key is used in securing links. Keys are deleted if their usage count exceeds a predefined threshold (uc). This step is executed regardless of the key transfer decision, After this step, if the node is going to transfer new keys and if it does not have enough space in its key ring, then it deletes some of its existing keys starting with the earliest used ones, until enough space is created for new keys. Key transfer operation is performed after the key deletion process. Hence, key ring size can never exceed m. The steps 11 and 12.a.i in our pseudo code corresponds to this key deletion process.

E. Performing Key Transfer

When a sensor node wants to request the keys in its T_i list from the base station, the node encrypts the requested key IDs with key K_{i-BS} and sends this message to the base station. Base station sends these keys to sensor node again by encrypting them with key K_{i-BS}. The number of keys that a sensor node can request from the base station at one time cannot exceed the Maximum Transfer Count (t_{max}). This part corresponds to the step 12.a.ii in our pseudo code.

After the key transfer operation is performed, or the keys in T_i list are added to the R_i list; node prepares itself for the next shared key discovery phase by clearing the C_i and T_i lists.

The main assumptions of this mechanism are as follows. Base stations are tamper-proof and they cannot be captured by an attacker. In addition, we assumed that each node can directly communicate with a base station in its communication range.

These assumptions require a powerful base station with high memory capacity and large communication range. The number of base stations needed depends on the wireless communication range of the base stations and the area of the deployment zone.

1- Nodes and base stations are pre-distributed with keys. Then they are deployed to the deployment area.
2- During the movement of nodes, the following steps are executed periodically:
  3- Shared key discovery phase is performed.
  4- Sensor node pairs, who share at least q common keys, establish a secure communication using all their shared keys.
  5- For each node n_i;
    6- The Most Frequent Keys list (C_i) is formed and sorted in decreasing order.
    7- Starting with the first key in C_i, tc number of keys are added to Key Transfer List (T_i), each with a probability of p.
    8- T_i list is sent to neighboring nodes and their lists are received.
    9- According to the T lists coming from neighbors and the Remembered Keys list (R_i), T_i list is updated.
    10- If the size of T_i list is greater than t_{max}, some of the keys in T_i list are deleted randomly, until the size of the list becomes equal to t_{max}.
    11- Keys that exceed the usage count (uc) are deleted from key ring.
    12- Node connectivity is calculated.
      12.a- If node connectivity is below the nc threshold, the keys in T_i list will be transferred;
        12.a.i- If there is not enough space in key ring for the transfer of new keys, some of the current keys are deleted, starting with the earliest used ones.
        12.a.ii- The keys in T_i list are transferred from the Base Station.
      12.b- If node connectivity is above the nc threshold, the keys in T_i list are added to the R_i list. If the size of R_i list becomes greater than rc, the oldest keys in R_i list are deleted, until enough space is opened for the latest remembered keys.
    13- C_i and T_i lists are cleared.

Fig 1. Pseudo code for DKRU mechanism

V. Performance Evaluation

The performance of our mechanism is evaluated via simulations, using C# for code development. A comparative analysis of Du's scheme with and without Dynamic Key Ring Update (DKRU) mechanism is given in subsections. Both random walk and RPGM mobility models are evaluated separately in each subsection. The common parameters and system configuration are as follows. Additional parameters are given in Table II.

- The number of sensor nodes in the network is 10,000.
- Deployment area is $1{,}000 \times 1{,}000$ square meters.
- Deployment area is divided into a grid of 10×10 cells; each cell has a group of 100 nodes in initial deployment.
- Size of the global key pool is 100,000.
- Size of the key pool for each group of nodes is 1789.
- Two horizontally and vertically neighboring key pools share exactly 0.2×1789 keys.
- Two diagonally neighboring key pools share exactly 0.05×1789 keys.
- Two non-neighboring key pools share no keys.
- Wireless communication range of sensor nodes is 40 m.
- Nodes are deployed to the grid cells using two dimensional Gaussian distribution.
- For mobility models, minimum and maximum speeds of nodes are 5 and 15 meters/minute, respectively.

TABLE II. List of Other Parameters Used In Simulations

	Du's scheme	**DKRU with RPGM**	**DKRU with random walk**
m	300	300	300
tc	-	3	3
p	-	0.6	0.6
t_{max}	-	10	10
nc	-	0.9	0.9
rc	-	80	80
uc	-	200	150

A. Global Connectivity Analysis

WSNs can also be viewed as key-sharing graphs where nodes are the vertices and secure links are the edges. Global connectivity is defined as the ratio of the size of the largest isolated component in this graph to the size of the whole network [4]. Nodes that are not connected to largest isolated component are considered as disconnected from the secure network. Hence, it is important to have high global connectivity in a network.

When global connectivity of Du's scheme is examined for RPGM model, it can be seen that even the network is fully connected at the beginning, in a short amount of time, only 10% of the network remains connected. This major decline results from the fact that two non-neighboring key pools do not share any keys in Du's scheme. When the initially non-neighboring groups become neighbors due to mobility, they cannot communicate and each group forms its own isolated component, which constitutes only 10% of the network. Our mechanism fixes this issue because nodes update their key rings according to their new neighbors. In Fig. 2, it can be seen that our mechanism provides almost perfect network connectivity for RPGM model.

In the random walk mobility model, global connectivity does not decrease significantly for Du's scheme. The reason is that because each node selects a new direction and speed periodically and randomly, they mostly stay in the same neighborhood. Their neighborhood consists of the vertically, horizontally and diagonally neighboring grid cells. Because the key pools of these cells have some overlapping, nodes can continue to establish secure links between them. As shown in Fig. 3, our mechanism also provides almost perfect global connectivity for this mobility model.

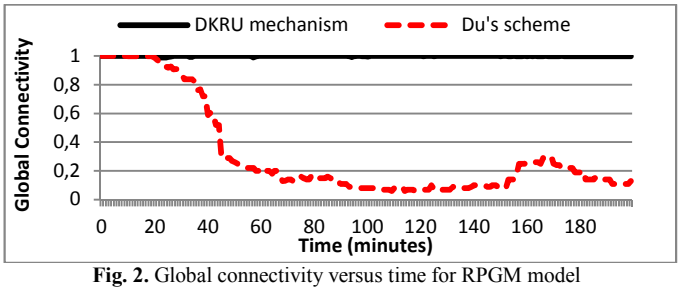

Fig. 2. Global connectivity versus time for RPGM model

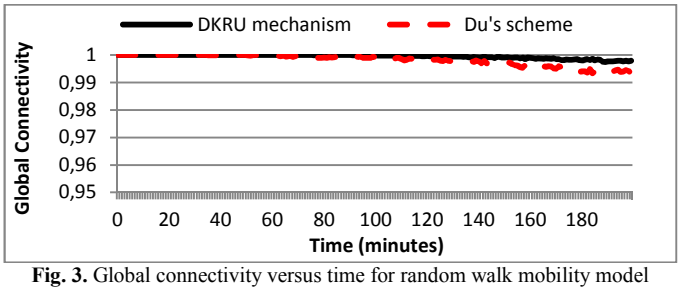

Fig. 3. Global connectivity versus time for random walk mobility model

B. Local Connectivity Analysis

We define local connectivity as the probability of two neighboring nodes being able to find at least 2 common keys to establish a secure communication link between them. Path key establishment phase is not taken into consideration in the computation of local connectivity, due to its high communication overhead. Hence, it is important for a network to achieve good local connectivity using shared key discovery phase alone. Moreover, path key establishment phase should be avoided in MWSNs because it involves much more communication and computational overheads compared to static WSNs [10].

Despite the fact that location based key distribution schemes provide better local connectivity than the probabilistic schemes for static WSNs, same situation is not valid for MWSNs. In Fig. 4 and 5, it can be seen that the local connectivity of Du's scheme decreases from 90% to 30% over time for both mobility models. This decrease is sharper for RPGM model because neighboring relationships break off faster in this model. When the DKRU mechanism is added to Du's scheme, local connectivity can be improved to 60% in steady state, without requiring any increase in key ring size.

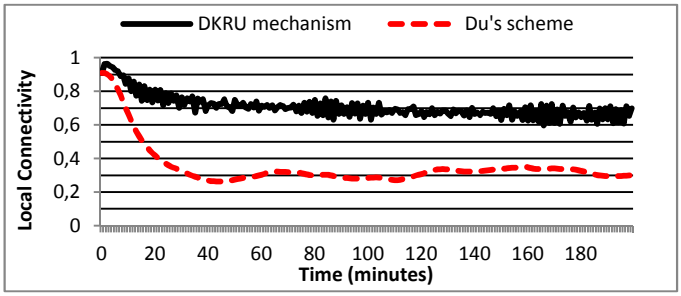

Fig. 4. Local connectivity versus time for RPGM model

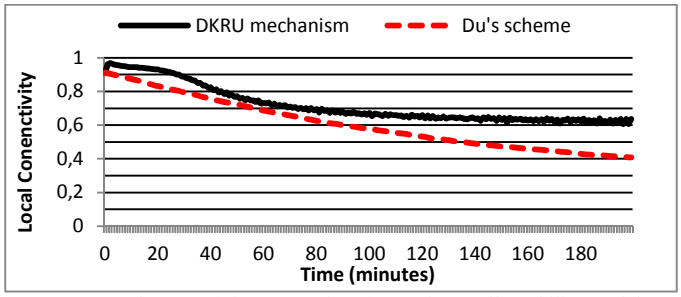

Fig. 5. Local connectivity versus time for random walk mobility model

C. Resiliency Analysis

One of the most important security threats for WSNs is the physical capture of sensor nodes by an attacker. Because the sensor nodes are not tamper proof, the attacker can access the key rings of sensor nodes and decrypt their communication. Moreover, these compromised keys may be used in communication links of non-captured nodes, too. In this case, the attacker can also decrypt the communications among non-captured nodes.

Resiliency of a network is inversely proportional to the amount of compromised links between non-captured nodes. In resiliency analysis, it is assumed that when an attacker captures a node, it retrieves all the keys in the node's key ring. Also, attacker has the ability to eavesdrop all message exchanges in the network. However, our attack model does not involve an active attacker who manipulates captured nodes to do further actions. In our simulations, attacker captures one node in each minute. Then we compute the ratio of additionally compromised links due to these node captures.

For the RPGM model, Du's scheme has very low global and local connectivity. Due to this low connectivity of network, it is hard to make judgments about the resiliency of network. As shown in Fig. 6, additionally compromised links ratio for Du's scheme is close to zero after 200 minutes of simulation. However, this does not indicate that the network is resilient. Actually, this indicates that there are not enough links in the network to be compromised. On the other hand, our mechanism provides high local and global connectivity for this mobility model. Despite this high connectivity, additionally compromised links ratio reaches only to 0.1 in our mechanism. This means, about 90% of the communication links between non-captured nodes are still secure.

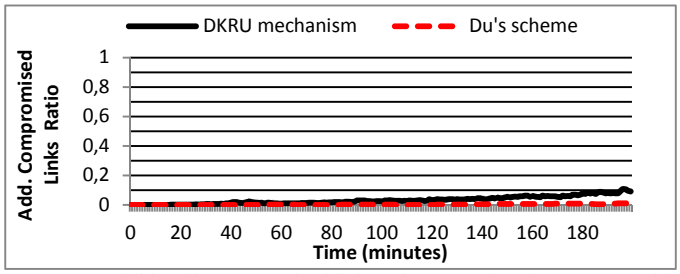

Fig. 6. Additionally compromised links ratio versus time for RPGM model

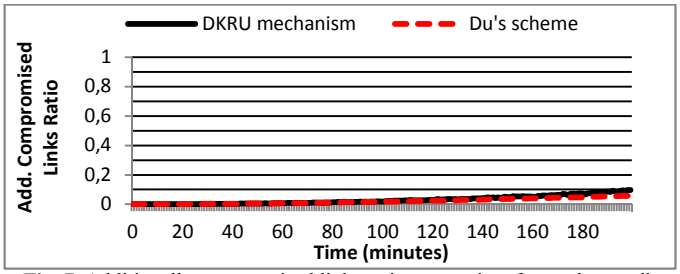

Fig. 7. Additionally compromised links ratio versus time for random walk mobility model

For the random walk mobility model, Fig. 7 shows that the additionally compromised links ratio of our mechanism is almost equal to the Du's scheme, except for a slight increase towards the end of the simulation. These results demonstrate that application of our mechanism does not significantly deteriorate the resiliency of network.

D. Communication Overhead Analysis

Communication overhead can be defined as the average number of bytes sent and received by a node at each shared key discovery phase. Without the DKRU mechanism, a node sends/receives all of the key IDs to/from its neighbors for shared key discovery phase. However, using DKRU mechanism results in additional communications. Firstly, nodes send the key IDs in their initial Key Transfer Lists to their neighbors and receive the key IDs from their neighbors. tc parameter affects the communication overhead of this step. Secondly, if a node is going to perform key transfer, it sends the requested key IDs to the base station and receives the encrypted keys. t_{max} parameter is important here because it determines how many keys will be requested from base station. In our computations, we considered 4-byte key IDs and 32-byte keys. As it can be seen in Fig. 8 and 9, communication overhead of our mechanism is very close to the communication overhead of Du's scheme. The reason is that, tc and t_{max} parameters do not require high values.

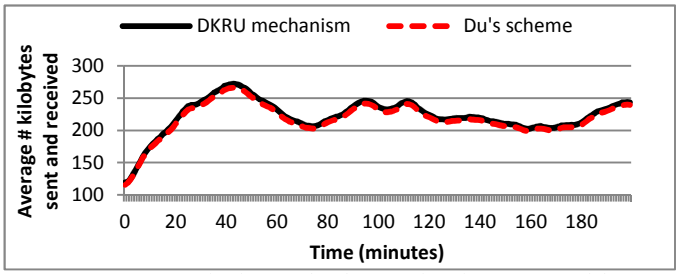

Fig. 8. Communication overhead versus time for RPGM model

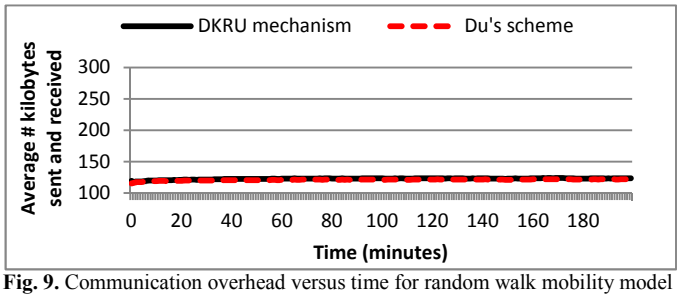

Fig. 9. Communication overhead versus time for random walk mobility model

VI. CONCLUSIONS

In this paper, we proposed Dynamic Key Ring Update (DKRU) mechanism for MWSNs. Our mechanism can be used together with different key pre-distribution schemes and it increases the local and global connectivity values of these schemes. Due to the mobile nature of network, neighbors of a node change continuously. Yet, DKRU mechanism helps sensor nodes to adapt to the network, regardless of their pre-deployment key distribution model. Using DKRU mechanism, a sensor node can update its key ring by observing the most frequent keys in its 1-hop and 2-hop neighbors' key rings. In this study, we explained our mechanism and we analyzed its performance when it is used together with Du's scheme, comparatively. We used two different mobility models for performance analysis. Our simulations show that DKRU mechanism provides a significant increase to local and global connectivity of Du's scheme in mobile case. Moreover, our mechanism does not cause an important decrease in the resiliency of network.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless sensor networks: a survey," Computer Networks, 38(4) pp. 393–422, 2002.

[2] L. Eschenauer, V. Gligor, "A key-management scheme for distributed sensor networks," in Proceedings of the Ninth ACM Conference on Computer and Communications Security (CCS'02), ACM, New York, NY, USA, pp. 41–47, 2002.

[3] Y. Zhou, Y. Fang, Y. Zhang, "Securing wireless sensor networks: a survey," Communications Surveys & Tutorials, IEEE , vol.10, no.3, pp. 6-28, Third Quarter, 2008.

[4] W. Du, J. Deng, Y. Han, S. Chen, P. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in Proceedings of the IEEE Computer and Communications Societies (INFOCOM'04), Los Alamitos, CA, USA, pp. 586–597, 2004.

[5] C. Haowen, A. Perrig, D. Song, "Random key predistribution schemes for sensor networks," Security and Privacy, 2003 Symposium on , vol., no., pp. 197- 213, 11-14 May 2003.

[6] S. Zhu, S. Xu, S. Setia, S. Jajodia, "Establishing pairwise keys for secure communication in ad hoc networks: a probabilistic approach," in 11th IEEE International Conference on Network Protocols (ICNP'03), 2003.

[7] K. Karaca, "A key distribution scheme tailored for mobile wireless sensor networks," Unpublished MS Thesis, Sabancı University, 2011.

[8] L. Zhou, J. Ni, C.V. Ravishankar, "Efficient key establishment for group-based wireless sensor deployments," in Proceedings of the 4th ACM Workshop on Wireless Security (Cologne, Germany, September 02 - 02, 2005). WiSe '05. ACM, New York, NY, pp. 1-10, 2005.

[9] A. Ünlü, A. Levi, "Two-tier, scalable and highly resilient key predistribution scheme for location-aware wireless sensor network deployments," Mob. Netw. Appl. 15, 4, pp. 517-529, August 2010.

[10] A. Kumar Das, "A Key Establishment Scheme for Mobile Wireless Sensor Networks Using Post-Deployment Knowledge," published in International Journal of Computer Networks & Communications (IJCNC) Vol.3, No.4, July 2011.

[11] K. Karaca, A. Levi, "Resilient key establishment for mobile sensor networks," Distributed Computing in Sensor Systems and Workshops, International Conference on, pp. 1-6, 2011.

[12] T. Camp, J. Boleng, V. Davies, "A survey of mobility models for ad hoc network research," Wireless Communications and Mobile Computing 2, pp. 483-502, 2002.